TP d'initiation à MATLAB n^0 1 : Première approche et généralités

Premier contact avec le logiciel MATLAB : on propose ici une suite de commandes à taper dans la fenêtre de commandes MATLAB. Après chaque ligne, taper la touche ↩ (*Entrée*). Attention, en programmation MATLAB, les minuscules et les majuscules sont des caractères différents.
L'objectif de ce TP est vous familiariser avec le langage MATLAB, sa structure, ses objets et ses fonctions, ainsi que d'avoir un apperçu de ses possibilités graphiques et des commandes Entréee/Sorties.

Avant de commencer

helpwin Ouvre la fenêtre d'aide

> La fenêtre d'aide présente d'abord les différents champs regroupant les commandes MATLAB. Avec la souris, choisir un des champs (par ex. *elmat*) par un double clic. Puis, choisir de la même manière une des commandes (par ex. *length*). Apparaît alors la description de la commande *length*. Revenir dans la fenêtre où vous avez taper *helpwin* et taper *help length*: la description est ainsi directement accessible. Faire d'autres essais. Pour terminer, iconifier la fenêtre d'aide graphique.

Commandes de type calculatrice

$6 + 3.4$	Effectue l'opération demandée.
$x = -2.34/21$	Le résultat de l'opération est affecté à la variable x.
x	Permet d'afficher la variable x.
$Y = x * 3;$	Création de Y. **Important:** terminer la commande par ; supprime l'affichage du résultat.
Y	Affiche la valeur de Y.
$Y = 1 + Y\hat{}3$	Opération de mise à une certaine puissance; remarquer que Y a changé, sa première écriture est oubliée. **On utilisera très souvent cette réaffectation d'une variable**.
who	Liste les objets MATLAB utilisés dans la session.
$JOIE = x + Y$	Affiche le résultat de $x + Y$ affecté à la variable $JOIE$.
who	Vérifier que $JOIE$ a été rajoutée. Noter que l'on peut appeler les variables comme on le veut. En pratique, on utilisera pour noms des variables les abréviations de ce qu'elles désignent. Par exemple, le temps pourra être t, T ou *temps*.
clear	Permet d'effacer toutes les variables de la session.
$x = 3.1;$	Nouvelle variable x.
$y = cos(x)$	Utilisation de la fonction cosinus.

> Si vous voulez rappeler une commande précedemment taper, ici par exemple *who*, il vous suffit d'utiliser les flèches ↑ (revient aux commandes précédentes) et ↓ (affiche les commandes suivantes). Faire plusieurs essais pour maîtriser ces commandes très utiles (évite surtout de recopier).

$z = exp(y)$ Utilisation de la fonction exponentielle.
Aller dans la fenêtre de help et dans le répertoire *elfun*; essayer ainsi les fonctions proposées.
On utilisera notamment souvent: *log*, *sqrt*, *abs*.

Programmes: boucles, conditionnement et M-fichiers

On va maintenant explorer les commandes principales permettant de réaliser des programmes MATLAB. Nous allons nous restreindre à des programmes effectuant des boucles avec de possibles conditionnements (nous verrons plus tard l'utilisation de sous-programmes et fonctions).

Premier programme: termes d'une suite

$clear$	On initialise. Important pour ne pas utiliser des variables déjà affectées.
$x = sqrt(2)/2;$	On initialise la variable du programme.
$for\ i = 1 : 100$	On va répéter 100 fois les commandes suivantes.
$x = 4 * x * (1 - x)$	On itère sur x la fonction logistique: on obtiendra donc une suite $u_{n+1} = 4u_n(1 - u_n)$.
end	On termine la boucle. S'affichent les 100 premiers termes de la suite.
i	Pour vérifier.

On recommence en ne s'arrétant que lorsque x est supérieur à 0.999

$clear$	On initialise.
$x = sqrt(2)/2;$	On initialise la variable du programme.
$i = 1;$	On utilise un compteur i pour connaître le nombre de termes nécessaires.
$while\ (x < 0.999)$	Boucle conditionnée.
$x = 4 * x * (1 - x);$	La suite "déroule".
$i = i + 1;$	Le compteur tourne.
end	Fin de la boucle conditionnée. Noter le résultat.
	Recommencer en changeant le conditionnement, puis en changeant le premier terme.

Nouveau programme avec test

$clear$	Nouveau programme.
$N = 247$	Initialisation de la variable
$i = 1;$	Compteur.
$while\ (N > 1)$	Boucle conditionnée.
$if\ (mod(N, 2) == 0)$	On conditionne suivant la parité de N. Noter que $=$ devient $==$ dans un test.
$N = N/2$	
$elseN = 3 * N + 1$	Donc dans le cas où N est impair.
$i = i + 1;$	Le compteur tourne.
end	Pour clore le test.
end	Pour clore la boucle. Recommencer en changeant de N.
	Vérifier que l'algorithme sur N le ramène toujours à 1 (ce que l'on ne sait pas montrer).

Recommencer à taper toutes les commandes pour changer seulement la valeur initiale n'est pas très pratique. Aussi utilise-t-on des fichiers de commandes que MATLAB reconnaîtra quand il se termine par *.m* (par ex. le fichier *Suite.m*). Pour cela, aller avec la souris dans *file*, puis *new*, écriver dans le fichier les commandes précédentes les unes sous les autres, puis sauvegarder avec *save*: mettre le nom de fichier *Suite.m*. Vous pouvez alors revenir dans la fenêtre principale MATLAB et en tapant *Suite* le programme est exécuté. Pour rappeler le fichier *Suite.m* (pour le modifier par exemple), aller avec la souris dans *file*, puis *open*.

Relancer plusieurs fois le programme *Suite* en changeant N (en en-tête du programme, il est important de mettre des commentaires pour permettre la reconnaissance. Pour cela, il suffit de commencer la ligne par le signe %. On peut aussi mettre des commentaires en cours de programme).
Faire une boucle sur N pour tester tous les nombres de 1 à 1000 (par exemple).

Vecteurs, graphiques et Entrées/Sorties

MATLAB (Matrix Laboratory) fonctionne essentiellement avec des vecteurs et des matrices. On commence par créer et utiliser des vecteurs. Pour illustrer des rpogrammes, il est important d'utiliser des représentations graphiques, de savoir les sauvegarder ainsi que des données et de les imprimer; c'est ce que nous verrons par la suite.

	Vecteurs
$X = 1:50$	On crée le vecteur X (noter l'emploi fréquent de majuscules pour désigner vecteurs et matrices).
$Y = 1:2:100$	Le vecteur Y contient les entiers $1, 3, .., 99$, construit avec le pas 2.
$Y(12)$	Donne la douzième composante de Y.
$Z = Y(4:17)$	Z est composé de la 4ième à la 17ième coordonnées de Y.
$length(Z)$	Taille du vecteur
$X + Y$	On peut sommer deux vecteurs de même taille. Faire la même chose avec $(-, .*, ./, .\hat{})$
$X + Z$	Recommencer les opérations pécèdentes. Conclusion?
$T = [3\ 2.5\ \ -6]$	Création d'un vecteur quelconque (espace ou , entre les nombres).
$save\ Vect\ T\ X$	On sauvegarde dans un fichier que l'on appelle par ex. $Vect$ les vecteurs T et X. Le fichier s'appelera $Vect.mat$.
$clear\ T\ X$	On efface ces deux vecteurs.
who	On vérifie.
$load\ Vect$	On "charge" le fichier $Vect.mat$; vérifier que T et X ont été affectés.
	Graphiques
$plot(X, Y)$	Trace les 50 points de coordonnées $(X(1), Y(1))$,..,$(X(50), Y(50))$, et les relie dans l'ordre.
$plot(X, Y,' c*')$	Impression sans relier les points, remplacés par $*$, et en couleur $cyan$. Noter que la nouvelle figure efface l'ancienne.
$hold$	Permet justement de conserver les anciennes figures.
$plot(X, 5 * sqrt(X),' -.')$	Nouveau tracé.
$title('Deux\ courbes')$	Met un titre à la figure
$xlabel('Valeurs\ de\ X')$	Axe des abscisses.
$figure(2)$	On crée une nouvelle feuille graphique.
$plot3(X, 2 * cos(X), 5 * sin(X))$	Tracé en 3 dimensions d'une courbe.

Pour conserver ce graphe, aller avec la souris dans *file*, puis *save as* (ou alors directement sur l'icone d'une disquette) et donner un nom (par exemple, *moche*). La figure sera conservée comme fichier *moche.fig*, et il suffira d'aller dans *open* pour la refaire apparaître dans une autre feuille graphique. On peut aussi imprimer la figure en allant avec la souris dans *file*, puis *print* (ou alors directement sur l'icone d'une imprimante), puis donner un nom et choisir soit une impression directe (*printer*), soit la sauvegarde comme fichier post-script, imprimable à loisir (*file*).

clf	Efface la figure(2).
$figure(1)$	Appelle la figure(1).
clf	Efface cette figure.
$N = input('Nombre\ N?\ ')$	Permet de choisir la valeur de la variable N. Insérer cette commande dans le fichier $Suite.m$ et faîtes tourner le programme plusieurs fois.

Exercice

1. Reprendre le calcul des termes de la suite "logistique" dans un M-fichier (*Suitelogistique.m*). Introduire le premier terme avec *input*, puis calculer les 501 premiers termes de la suite, mémorisés dans un vecteur X. Tracer les points $(X(n), X(n+1))$, pour $n = 1, 2..500$. Imprimer lorsque la figure est signifiante.